Module 7: Hands-On: Kubernetes Dashboard

Run the following commands for installing kubeadm as root (both master and worker)

sudo su

apt-get update

apt-get install docker.io

apt-get update && apt-get install -y apt-transport-https curl

curl -s https://packages.cloud.google.com/apt/doc/apt-key.gpg | apt-key add -

cat <<EOF > /etc/apt/sources.list.d/kubernetes.list

deb https://apt.kubernetes.io/ kubernetes-xenial main

EOF

apt-get update

apt install kubeadm=1.21.0-00 kubectl=1.21.0-00 kubelet=1.21.0-00 -y

Creating cluster:

Initializing kubeadm on master using:

kubeadm init --pod-network-cidr=192.168.0.0/16

```
Your Kubernetes control-plane has initialized successfully!

To start using your cluster, you need to run the following as a regular user:

  mkdir -p $HOME/.kube
  sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config
  sudo chown $(id -u):$(id -g) $HOME/.kube/config

Alternatively, if you are the root user, you can run:

  export KUBECONFIG=/etc/kubernetes/admin.conf

You should now deploy a pod network to the cluster.
Run "kubectl apply -f [podnetwork].yaml" with one of the options listed at:
  https://kubernetes.io/docs/concepts/cluster-administration/addons/

Then you can join any number of worker nodes by running the following on each as root:

kubeadm join 172.31.23.197:6443 --token om12ul.nhq0adgipvqii3yg \
        --discovery-token-ca-cert-hash sha256:1665c8651063e91650362aa60aa89724d63c68397f34387b44e715d8
90ca6bca
root@ip-172-31-23-197:/home/ubuntu#
```


Copy the Join token and paste in the worker machine.

```
This node has joined the cluster:
* Certificate signing request was sent to apiserver and a response was received.
* The Kubelet was informed of the new secure connection details.

Run 'kubectl get nodes' on the control-plane to see this node join the cluster.
```

We need to run the below commands:

mkdir -p $HOME/.kube

sudo cp -i /etc/kubernetes/admin.conf $HOME/.kube/config

sudo chown $(id -u):$(id -g) $HOME/.kube/config

To list all nodes :

kubectl get nodes

```
root@ip-172-31-23-197:/home/ubuntu# kubectl get nodes
NAME                STATUS     ROLES                  AGE     VERSION
ip-172-31-23-197    NotReady   control-plane,master   4m44s   v1.21.0
ip-172-31-37-241    NotReady   <none>                 4m17s   v1.21.0
```

It shows the nodes but the status is not ready because we have not installed the network plugin.

To install network plugin, run the below commands:

kubectl apply -f https://docs.projectcalico.org/manifests/calico.yaml

kubectl apply -f https://raw.githubusercontent.com/kubernetes/ingress-nginx/controller-v0.49.0/deploy/static/provider/baremetal/deploy.yaml

Check the nodes for its state after installing network plugins.

kubectl get nodes

```
root@ip-172-31-23-197:/home/ubuntu# kubectl get nodes
NAME                STATUS   ROLES                  AGE     VERSION
ip-172-31-23-197    Ready    control-plane,master   6m54s   v1.21.0
ip-172-31-37-241    Ready    <none>                 6m27s   v1.21.0
```


Thus, we have successfully installed Kubernetes. Run the below command to create a dashboard:

kubectl apply -f https://raw.githubusercontent.com/kubernetes/dashboard/v2.0.0/aio/deploy/recommended.yaml

Then edit the service:

kubectl edit service kubernetes-dashboard -n kubernetes-dashboard

```
# Please edit the object below. Lines beginning with a '#' will be ignored,
# and an empty file will abort the edit. If an error occurs while saving this file will be
# reopened with the relevant failures.
#
apiVersion: v1
kind: Service
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"k8s-app":
s-dashboard"},"spec":{"ports":[{"port":443,"targetPort":8443}],"selector":{"k8s-app":"kuber
  creationTimestamp: "2022-01-13T07:16:45Z"
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard
  namespace: kubernetes-dashboard
  resourceVersion: "1539"
  uid: 01f6f13a-d1bf-45b0-afc4-ca9713168cac
spec:
  clusterIP: 10.105.231.218
  clusterIPs:
  - 10.105.231.218
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - port: 443
    protocol: TCP
    targetPort: 8443
  selector:
    k8s-app: kubernetes-dashboard
  sessionAffinity: None
  type: ClusterIP
status:
```

Change ClusterIP to NodePort

Note: We need to change the type from ClusterIP to NodePort. The below given image contains the modified service file.

```
apiVersion: v1
kind: Service
metadata:
  annotations:
    kubectl.kubernetes.io/last-applied-configuration: |
      {"apiVersion":"v1","kind":"Service","metadata":{"annotations":{},"labels":{"k8s-app":
s-dashboard"},"spec":{"ports":[{"port":443,"targetPort":8443}],"selector":{"k8s-app":"kuber
  creationTimestamp: "2022-01-13T07:16:45Z"
  labels:
    k8s-app: kubernetes-dashboard
  name: kubernetes-dashboard
  namespace: kubernetes-dashboard
  resourceVersion: "1997"
  uid: 01f6f13a-d1bf-45b0-afc4-ca9713168cac
spec:
  clusterIP: 10.105.231.218
  clusterIPs:
  - 10.105.231.218
  externalTrafficPolicy: Cluster
  ipFamilies:
  - IPv4
  ipFamilyPolicy: SingleStack
  ports:
  - nodePort: 31707
    port: 443
    protocol: TCP
    targetPort: 8443
  selector:
    k8s-app: kubernetes-dashboard
  sessionAffinity: None
  type: NodePort
status:
  loadBalancer: {}
```

Note: The editor used is vim. Therefore to save and exit we need to press ESC and then: wq

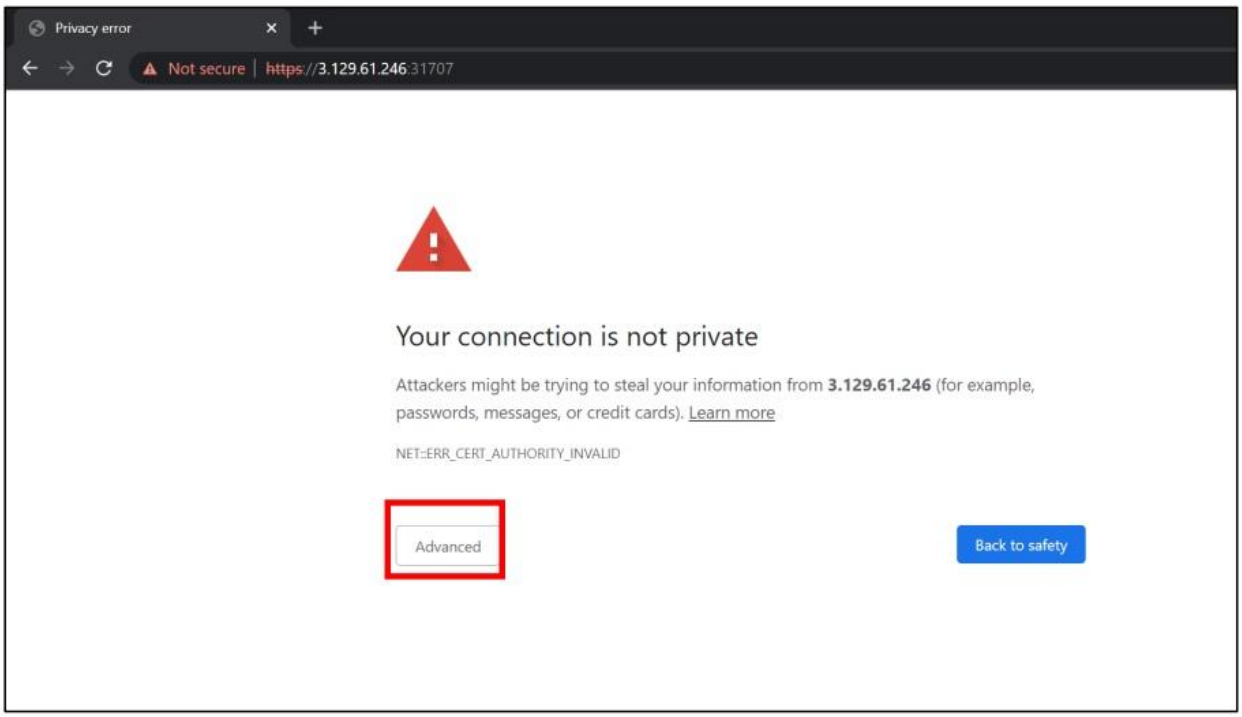
```
root@ip-172-31-23-197:/home/ubuntu# kubectl edit service kubernetes-dashboard -n kubernetes-dashboard
service/kubernetes-dashboard edited
```

kubectl get svc -n kubernetes-dashboard

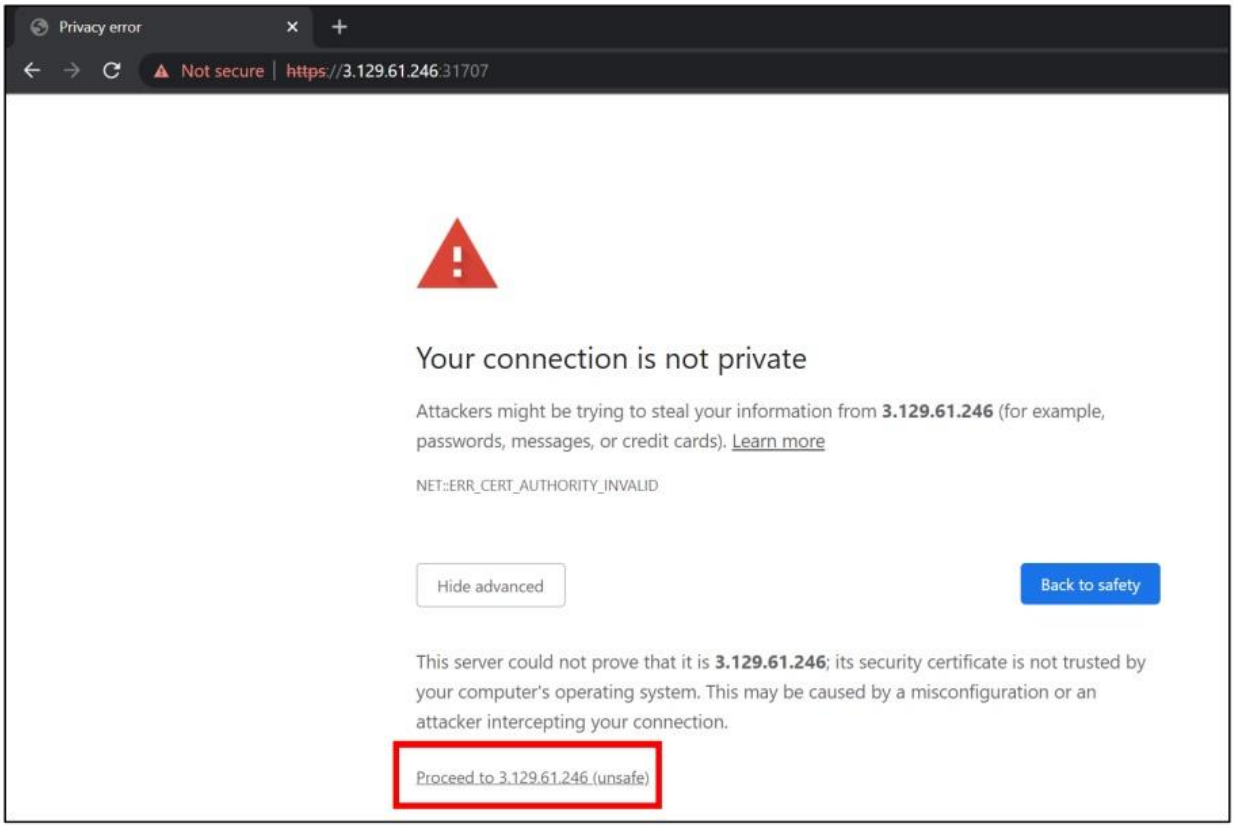
```
root@ip-172-31-23-197:/home/ubuntu# kubectl get svc -n kubernetes-dashboard
NAME                        TYPE        CLUSTER-IP       EXTERNAL-IP   PORT(S)         AGE
dashboard-metrics-scraper   ClusterIP   10.108.75.42     <none>        8000/TCP        16m
kubernetes-dashboard        NodePort    10.105.231.218   <none>        443:31707/TCP   16m
root@ip-172-31-23-197:/home/ubuntu#
```

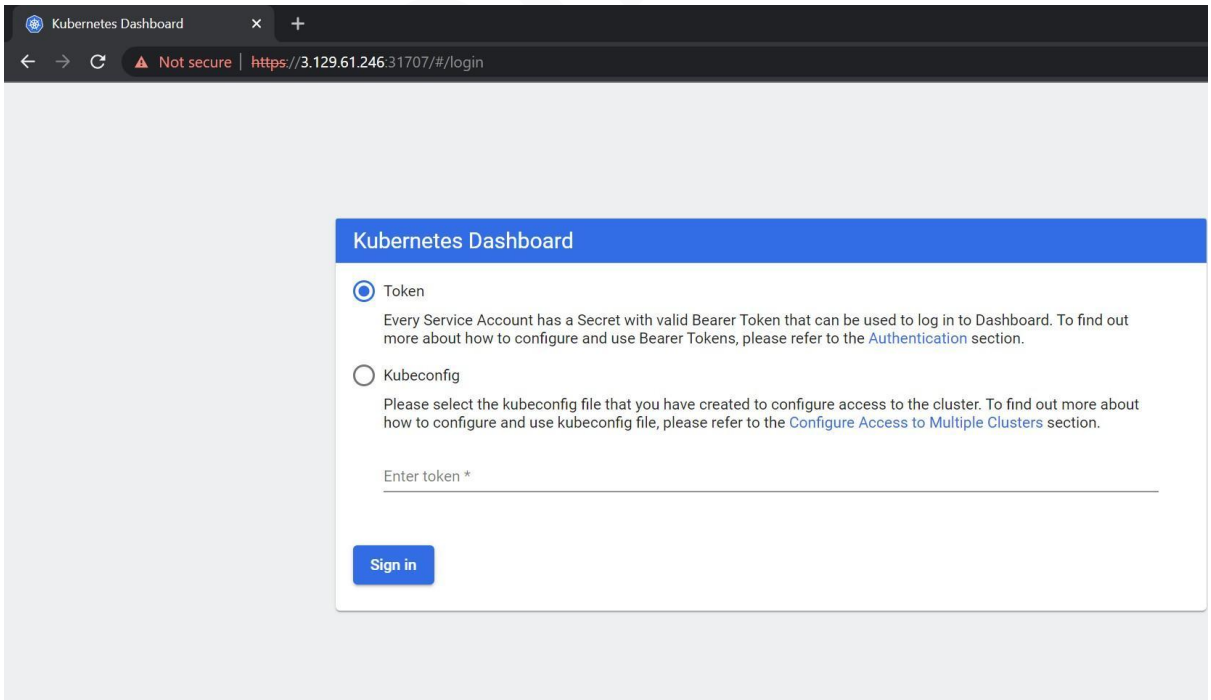

Now go to a browser and paste the ip along with the port
https://<ip-of-master> :31707

We need to click on Advance option to access the webpage



Click on the Proceed option to open the dashboard





We need to create a service account:

To create service account run the below command:

```
kubectl create serviceaccount cluster-admin-dashboard-sa
```

To bind clusterAdmin role to the service account use the below command:

```
kubectl create clusterrolebinding cluster-admin-dashboard-sa \
 --clusterrole=cluster-admin \
 --serviceaccount=default:cluster-admin-dashboard-sa
```

To parse the token run the below command:

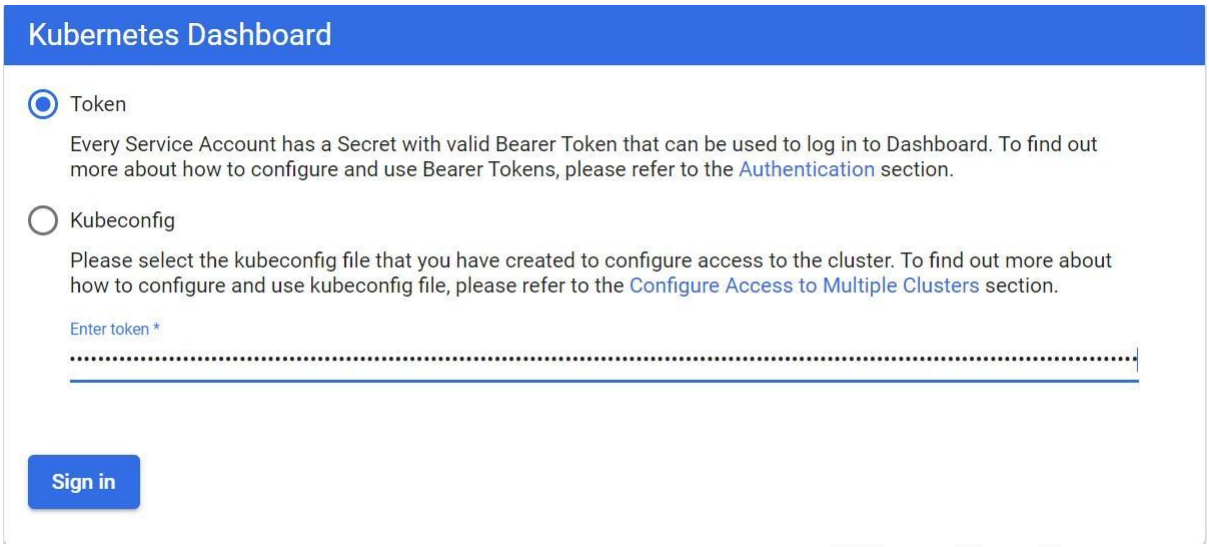
```
TOKEN=$(kubectl describe secret $(kubectl -n kube-system get secret | awk '/^cluster-admin-dashboardsa-token-/{print $1}') | awk '$1=="token:"{print $2}')
```

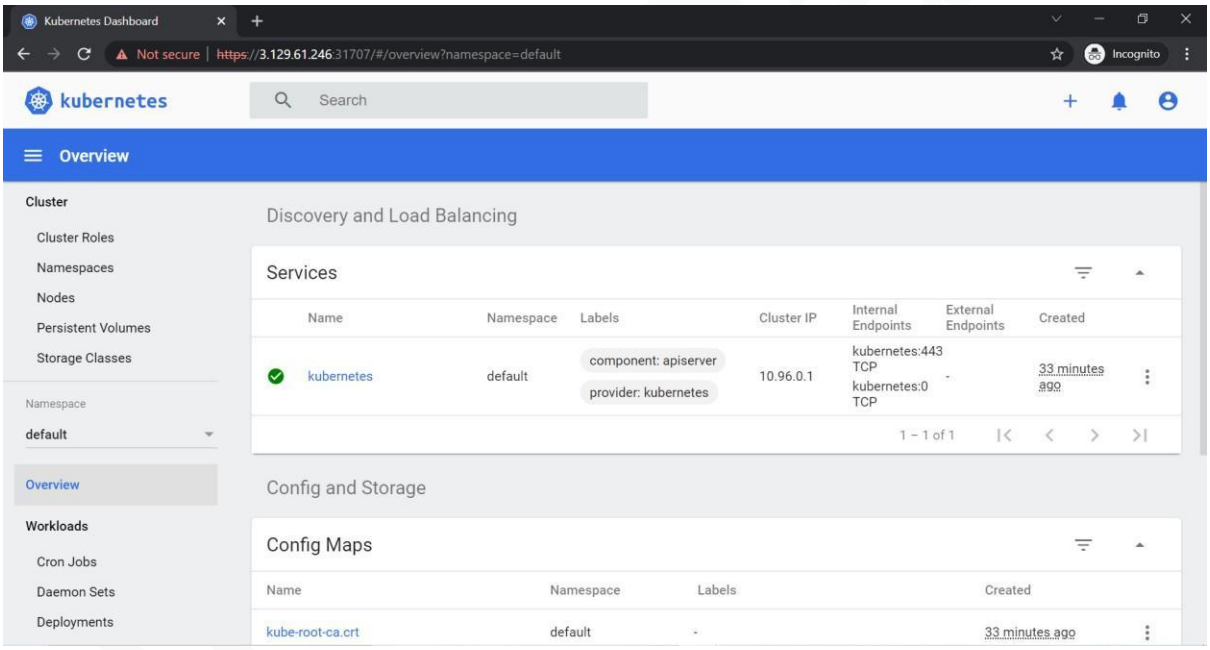
Then we need to run the below command:

```
echo $TOKEN
```

```
root@ip-172-31-23-197:/home/ubuntu# echo $TOKEN
eyJhbGciOiJSUzI1NiIsImtpZCI6Ikx0MzdxclRxZG1zcWhoNnNMTm5yTXN6MEZ5ZGRyT3dCc0dSVk1uSS1UVkEifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZ
ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJkZWZhdWx0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZWNyZXQubmFtZSI6
tZGFzaGJvYXJkLXNhLXRva2VuLWc2cmp6Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQubmFtZSI6ImNsdXN0ZXItYWRta
Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZXJ2aWNlLWFjY291bnQudWlkIjoiYWM2MmMzNTItMzA3NC00ZWExLThlMzUtODM1Njg0YWZkZTQ1Ii
nNlcnZpY2VhY2NvdW50OmRlZmF1bHQ6Y2x1c3Rlci1hZG1pbi1kYXNoYm9hcmQtc2EifQ.m-m9wtaRNgJb74CN-d20RD7BsCUvlh8KeAq-H_TobJf-oGvpPJxd
CFIJQRM6CywQvH3yIoiw0kUrMXUwkZbT9LO30ap-_kpTiQcu28SsKPhb8JOltFIBdAvC6Zxo_2WUlYirtM1pscjlwQQNURhht8vU3YKO06vCC6yK8TNOCBgjym
xDSO8ORTDmm4JQ8_Tzy17ZhlDzsWWBARB_3m7kas2m_ZXgpJ-vUgL4XdOPg5qIei6BleE8UDDbOV-PKY4hfJZ17t483piQm3CxQ_-uVS_DsS9P_o5h3DevhSRI
JSUzI1NiIsImtpZCI6Ikx0MzdxclRxZG1zcWhoNnNMTm5yTXN6MEZ5ZGRyT3dCc0dSVk1uSS1UVkEifQ.eyJpc3MiOiJrdWJlcm5ldGVzL3NlcnZpY2VhY2Nvd
cy5pby9zZXJ2aWNlYWNjb3VudC9uYW1lc3BhY2UiOiJkZWZhdWx0Iiwia3ViZXJuZXRlcy5pby9zZXJ2aWNlYWNjb3VudC9zZWNyZXQubmFtZSI6ImRlZmF1bH
CJrdWJlcm5ldGVzLmlvL3NlcnZpY2VhY2NvdW50L3NlcnZpY2UtYWNjb3VudC5uYW1lIjoiZGVmYXVsdCIsImt1YmVybmV0ZXMuaW8vc2VydmljZWFjY291bnQ
50LnVpZCI6IjZjOWUyMDY4LWI4MzItNGMwOS04YWYyLWM5MWRiYjE1MDk4ZiIsInN1YiI6InN5c3RlbTpzZXJ2aWNlYWNjb3VudDpkZWZhdWx0OmRlZmF1bHQi
XCrZBlil4ztVArzK3Ymo3CsOCIH0rKtrQrxQoxHH8-k3Tj1_WYabVsm-ND-r8ewW398IghTZ1Mu_K9kXwCZ3XT--P6OeYZFYvCzkuuzqJ8cdji1i9dNXFMa6lb
FnV-0Xx7krrao_jiviIb7fUXw7cNp8rosVaH2u4tfbNXWNyTneG_KV1pY5G_nVY7l3iS3uXBfywR_8DkI8cEUaUNN1OePskKnJLGMRERnxPdH2EDUaaZ-8Kx4a
4blgHIQtjLwjwq0fywPR5r6RdCjh8X7c55DfYg5rG4Eg
root@ip-172-31-23-197:/home/ubuntu#
```

Copy the token and paste in the Kubernetes dashboard.





Then click on Sign in. The dashboard is created.

